TECHNICAL REPORT 4776

A COMPARISON OF METHODS FOR CALCULATING NON-TIME-DEPENDENT RELIABILITY

ARTHUR J. HEYDERMAN

JUNE 1975

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PICATINNY ARSENAL
DOVER, NEW JERSEY

The findings in this report are not to be construed as an official Department of the Army Position.

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
Technical Report 4776

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A COMPARISON OF METHODS FOR CALCULATING NON-TIME DEPENDENT RELIABILITY

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Arthur J. Heyderman

8. CONTRACT OR GRANT NUMBER(s)
AMCMS Code 664603.12.26300

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Product Assurance Directorate
Picatinny Arsenal
Dover, NJ 07801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
June 1975

13. NUMBER OF PAGES
25

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Nuclear
Reliability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A major interest exists in improved ways for calculating reliability point estimates for non-time-dependent single-shot devices such as those found in tactical nuclear weapons. Often decisions involving many millions of dollars and even human lives depend upon the accuracy of the models used and the estimates obtained. Seven methods are investigated which have been considered or used to calculate reliability estimates. Of those techniques

20. Continued

investigated, four are in use at the present time. These four, "SABRE", "GO", "SYSTEMEX" and Failure Equations are compared and an indication is made of their strong and weak points. No single method was found to be universally superior for all uses and indeed no single method has been universally accepted by all agencies dealing with tactical nuclear weapons. Moreover, the sensitive nature of these items has made the use of two independent analyses by separate techniques a very desirable cross-check on the analyst.

TABLE OF CONTENTS

	Page No.
Purpose	1
Introduction	1
Techniques Available	3
Sample Problem	4
QUEST	5
The Tree Method	7
Failure Equations	9
SYSTEMEX	10
GO	11
TRI-SERVICE Method	15
SABRE	16
Comparison of Techniques	16
Other Investigations	18
Conclusions	18
References	19
Distribution List	21

PURPOSE

The purpose of this paper is to examine various methods of calculating a reliability point estimate for non-time-dependent[1] single -shot[2] devices such as tactical nuclear weapons.

INTRODUCTION

Tactical nuclear weapons are an important element in this nation's arsenal. As such, a great deal of emphasis has been placed on high operational reliability should their use ever be required. Bear in mind that no tactical weapon has ever actually been used in war. Furthermore, the test ban treaty prohibits the atmospheric testing of a complete round,[3] including a full nuclear explosion. It then becomes apparent that all reliability data on these weapons must arise from test programs on component sub-systems and allied sources. This data must in turn be applied to models in order to provide system level reliability estimates from component level data. Realizing this, the importance of accurate and efficient analysis techniques is apparent. The high cost and politically sensitive nature of nuclear weapons have dictated that extremely high safety and reliability requirements be established requiring the use of extremely complex safing and arming systems. The cost of testing these systems is further increased by the severe storage and operating environments to which the test items must be exposed. This precludes the testing of a sufficient quantity of complete rounds to enable a meaningful analysis of the weapon as a whole. As a result methods have evolved to model systems based on components and to calculate system reliability point estimates from component data. Several organizations have developed their own preferred method of analysis. This paper will attempt to examine

[1] Non-time-dependent: The reliability of the device does not change throughout the duration of the mission. Although degradation over the life of the item may occur, estimates are made at one point in time.

[2] Single-shot device: Item cannot be reused.

[3] Unless indicated otherwise, the term "round" will be used to encompass all types of tactical nuclear devices regardless of actual delivery system. Artillery fired rounds will be termed "projectiles."

the various methods which exist and will provide comparisons. Items of consideration will include the delivery system[1] of the weapon and the various types of studies performed. These studies include feasibility and development, design verification, and stockpile surveillance programs. Each of these studies presents different problems. For feasibility studies and later on in development, data usually has not been accumulated on the various components. In this case reliability estimates are based on extremely detailed models and hypothetical component data. These models are in sharp contrast to the less detailed models found in the design verification studies which utilize values extrapolated from test data and hence tend to combine some components into "black boxes." Finally, the stockpile surveillance studies models remain unchanged from the design verification test models; however, the data used is generally based on small sample sizes[2] and many data points have zero failures, thus demonstrating the need for more detailed observations, especially on components with internal redundancies. Any comparison of analytic techniques must consider each of these different types of studies, and similarly each of the three types of delivery systems. For example, in the case of an artillery fired atomic projectile (AFAP) all data for verification and stockpile tests is collected by a firing program, which in this case is simpler than a laboratory program. As a result fewer components appear in the models due to telemetry limitations. Missile fired systems (adaption kits) have such enormous delivery vehicle costs that most of their testing is performed in the laboratory where much more voluminous and detailed data is available which then leads to a more detailed model. Finally, atomic demolition munitions (ADM's) generally have a much longer operating time and may be much more complicated than other weapons, this due to the requirement for manual emplacement and delayed detonation. The ease with which each of these systems can be analyzed will be an important factor in this comparison of the analysis techniques.

[1] Delivery system: The method of placing the weapon at the desired point of detonation. Typical delivery systems of tactical nuclear weapons are: artillery shell, missile and demolition charge.

[2] Small sample sizes for nuclear weapons are generally on the order of two to twenty items.

TECHNIQUES AVAILABLE

At the present time there are numerous methods for calculating system reliability point estimates. Several of these methods will be discussed. They will be referred to as: "QUEST" (Ref 1), the "tree" method (Ref 2), "failure equations," "SYSTEMEX" (Ref 3), "GO" (Ref 4), the "TRI-SERVICE" method (Ref 5), and "SABRE" (Ref 6). Each is briefly described below.

QUEST is a computerized Monte Carlo approach for calculating system point estimates from a Boolean success model.[1] It was developed by the Data Processing System Office of Picatinny Arsenal in 1966.

The "tree" method is a computerized approach to calculating a system point estimate from a Boolean success model by construction of a success-failure tree. This method was limited by the number of components in the system and never adopted for use. It was, however, a forerunner of other methods now in use. The tree method was developed by the Mathematics and Statistics Branch of the Nuclear Reliability Division at Picatinny Arsenal in 1967.

The failure equation method is a very simple technique for approximating the probability of failure of a high reliability system. This method uses a Boolean success model of the system and then considers only system failure caused by at most two independent component failures, the assumption being that third and higher order[2] system failures are negligible. Then, by summing the first and second order failures, a system probability of failure is computed and the approximate point estimate obtained. This method was developed by the Sandia Corporation for the Polaris missile and is still widely used today. Documentation is lacking, but the method is simple enough to allow a complete description to be presented.

[1] Boolean success model: A description of successful operation of a system in terms of unions and intersections of sets of component outcome events. Each component has two disjoint outcomes, i.e., success or failure and all components are independent of each other.

[2] Third order: The order of a failure is the number of independent components causing the failure.

SYSTEMEX is a computerized method for expanding Boolean success models into success equations. It is considered an exact equation by its users; however, precise would be a better description. This method was also developed by the Mathematics and Statistics Branch of the Nuclear Reliability Division, Picatinny Arsenal.

GO is a computerized tree method for calculating a system reliability point estimate from component data utilizing a sophisticated modeling technique developed as part of the GO approach. GO was developed in 1968 by Kaman Sciences Corporation, working under contract to Picatinny Arsenal on the Safeguard Anti-Ballistic Missile (ABM) System.

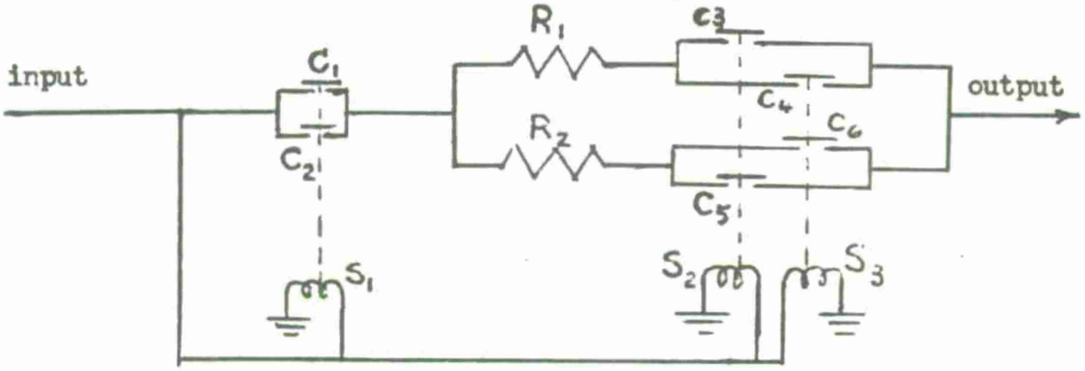
The TRI-SERVICE technique is a Monte Carlo approach to computing system reliability as a distributed variable rather than a point estimate. The model is similar to a Boolean success model; however, certain restrictions exist. This method was developed jointly in 1972 by Picatinny Arsenal, Kelly Air Force Base and the U.S. Naval Ammunition Depot, Oahu.

SABRE is an off-shoot of the TRI-SERVICE approach utilizing either the GO modeling technique or a failure equation. Like the TRI-SERVICE approach the output is a distributed variable rather than a point estimate. This method was developed at Picatinny Arsenal in 1973 and is coming into use on the newer weapons systems, especially artillery fired atomic projectiles.

SAMPLE PROBLEM

In order to compare the various techniques we will first pose a sample situation and then look at how each method would approach the problem.

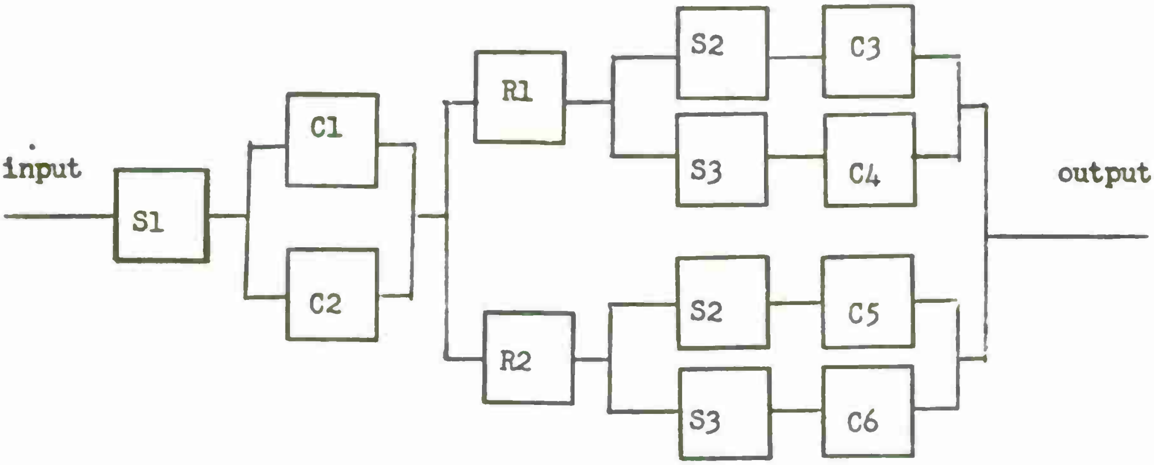
Consider the simple electro-mechanical system illustrated on the following page. The system contains three electrically operated relay switches, each of which closes two contacts. There are also two resistors in the system for a total of eleven (11) components. Like components are identical for the purposes of this example. Below the diagram are the original engineer estimates of the component reliabilities and the results of test data on twenty-one (21) flights of the system, expressed as failures out of numbers tested. For each of the techniques we will consider how a point estimate will be made from the engineer estimates and then consider what the effects of the real data would be on this original point estimate.



ENGINEER ESTIMATES	TEST DATA 21 FLIGHTS
R = .99	R - 0 of 42
C = .999	C - 1 of 126
S = .90	S - 4 of 63

QUEST

QUEST is a computer program utilizing a Monte Carlo technique to convert a Boolean success model into a point estimate of reliability. The first step then, in the solution of our sample problem, would be the representation of the system as a logical (Boolean) success model. This is done by considering each component in the system and determining the sequence of events which would lead to a successful system operation. In our sample problem we would have an input followed by a successful activation of the S1 relay closing either C1 or C2, permiting power to reach point A. At that time we need either of two identical channels to operate where a channel operation would consist of a successful resistor operation followed by one of two contacts being successfully closed by its respective relay. This explanation is represented pictorially on the following page by what is termed a block diagram. In this diagram each block represents the successful operation of the represented component. Any one minimal (i.e., no unnecessary blocks) path from input to output in the block diagram is considered sufficient for success and as such is termed a "success path" or "tie-set." We will also define a "cut-set" as a set of events (blocks) such that all success paths have at least one block in the "cut-set." Thus, a cut-set is a set of elements sufficient to preclude successful operation of the system.



The Boolean (logical) success model above is a common starting ground for many of the methods to be considered in this paper and is in fact sometimes a severe restriction on the usefulness of the approach. This will be discussed more fully later in the paper.

Returning to the procedures in QUEST we would now express system success in terms of the elementary success events using logical operators as follows:

$$\text{SYSTEM SUCCESS} = (S1\wedge(C1\vee C2))\wedge((R1\wedge((S2\wedge C3)\vee(S3\wedge C4)))\vee(R2\wedge((S2\wedge C5)\vee(S3\wedge C6))))$$

This would then be entered into the computer as a subroutine of QUEST and the program would perform a Monte Carlo simulation until satisfactory results were obtained.

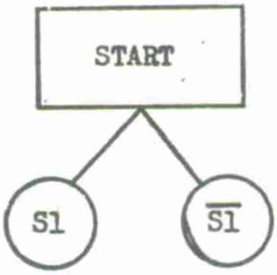
To perform one single simulation run the computer would select one number for each component from a uniform distribution on the closed interval (0, 1). Then for each component it compares the selected random number with the established reliability value and if the random number (considered stress) does not exceed the reliability value (considered strength) then a success is assigned to that component; otherwise the component (block) is assigned a failure. Having assigned a success (true) or failure (false) to each block in logical success equation above we would calculate the system success as either "true" or "false." By repeating this procedure a sufficient number of times, a ratio could be formed as the number of system successes divided by the number of trials. This ratio would be the point estimate of system reliability.

This procedure was actually performed for the sample system. Using the engineer estimates a reliability point estimate of 0.89 was obtained while the test data produced a result of 0.930 for the system.

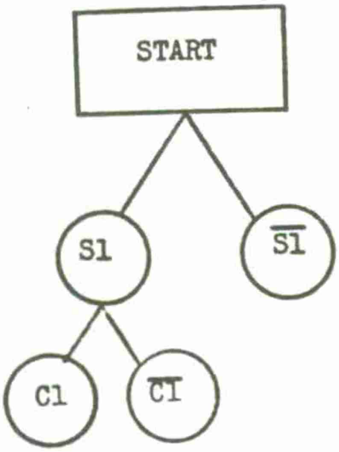
THE TREE METHOD

Like QUEST, the tree method is a computerized method for converting a Boolean success model into a point estimate of reliability. In addition, this program also has the capability of producing an algebraic equation for reliability in terms of the probabilities of success for each of the component events. This algebraic equation is termed a success equation.

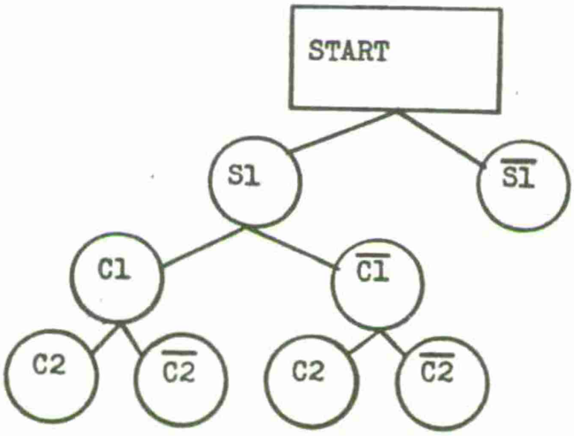
The program constructs a success-failure tree by sequentially entering each component event and branching for each state. For the sample problem we would start with two branches as shown.



Where $\overline{S1}$ represents the failure of the S1 relay. The next step would be to look at each branch and determine if it forms either a cut-set or success path. In this case $\overline{S1}$ is a cut-set. For those branches that form cut-sets or success paths no further branching is done. On all other branches the next component event is added as shown below.



In this case no new cut-sets or success paths have been produced so the program proceeds to branch on the next component as shown below.



At this point the branch S1, $\overline{C1}$, $\overline{C2}$ is also a cut-set and thus there would be no further branching below on that branch. The program continues with this procedure until all elements in the system are exhausted. At that point every branch in the tree is either a cut -set or a success path. A success equation is then formed as the sum of all the success paths in the tree and the probabilities are inserted in this equation to yield a point estimate. For even the sample system shown the procedure is too tedious to complete by hand. Furthermore, the program has been in a state of disuse for such a long period that it would be of little value to resurrect it merely to solve a simple example. The actual results produced by the tree program are identical to those produced by SYSTEMEX, namely a success equation and the precise point estimate resulting.

FAILURE EQUATIONS

Perhaps the conceptually simplest of all the procedures examined is the use of failure equations. In the failure equation method you merely sum the probabilities of all the "easy" failure modes and subtract the total from unity for a point estimate. By "easy" we mean cut-sets containing not more than two component events. Returning to our sample problem, we see that there are four possible ways by which the system can fail from two or less components. Either the S1 relay can fail to function, or both C1 and C2 contacts can fail to close, or both R1 and R2 resistors can fail (open circuit), or both the S2 and S3 relays can fail to function. This gives a failure equation

$$P(\text{failure}) = \bar{S} + \bar{C}^2 + \bar{R}^2 + \bar{S}^2$$

where the bar represents probability of failure. The point estimate for our sample system would be .890 using engineer estimates and .932 using the test data.

It is of interest in the discussion of failure equations to explore just which terms have been ignored so as to justify in some small way what at the outset might seem an outrageously inaccurate method. First there is the question of cut-sets which have not even been included such as ($\overline{S2}$, $\overline{C5}$, $\overline{R2}$) for example. In general there are a huge number of higher (than two) order cut-sets in comparison to the number involved in the failure equation. The assumption is that the sum of all these terms is negligible. For real weapons systems this assumption is in fact valid since typically each component reliability is of the order of .99 or .999. The sum total effect of all these higher order terms would be to slightly lower the estimate of reliability. To further compensate for this error, the method also ignores all intersections of those failure terms which were included. For example, in our sample problem we had the term $\overline{S1}$ and $\overline{C1}\ \overline{C2}$ but these counted the term $\overline{S1}\ \overline{C1}\ \overline{C2}$ twice and so one $\overline{S1}\ \overline{C1}\ \overline{C2}$ should have been subtracted. The net of all these error terms would be to slightly raise our estimate of reliability. Thus, two sets of very small error terms are both ignored in this method but these errors are both small and compensating with the resulting failure estimate being very precise. In fact, differences between estimates using failure methods and using other methods on real systems have proven negligible.

SYSTEMEX

There are several techniques for the generation of success equations. We have already looked at the tree method which has a limited capability to produce success equations. Two other programs of interest are SYSTEMEX and SYSTEMEQ (Ref 7).

These two procedures are virtually identical except that SYSTEMEQ automates some of the computations done by hand for SYSTEMEX. We will only consider SYSTEMEX, it being the original version. Before considering the actual operations of the program one point should be noted. The probability of success of the union of two events is the sum of the success probabilities of each event, minus the success probability of the intersection of the two events. Now if two events are independent, then the probability of success of the intersection is the product of probabilities of success of the two events. Thus for example $P(S1 + S2) = P(S1)+P(S2)-P(S1S2)=P(S1)+P(S2)-P(S1)P(S2)= 2S-S^2$ for S1 and S2 independent events of success probability S.

Returning now to the operation of the SYSTEMEX routine the first step is to construct the Boolean success model of the system. This procedure has been explained previously. Having constructed the model the next step is to list all the success paths individually. In the case of our sample problem we would have:

$$\begin{aligned} \text{SUCCESS} = {} & S1\ C1\ R1\ S2\ C3 + S1\ C1\ R1\ S3\ C4 \\ & + S1\ C2\ R1\ S2\ C3 + S1\ C2\ R1\ S3\ C4 + S1\ C1\ R2\ S2\ C5 \\ & + S1\ C1\ R2\ S3\ C6 + S1\ C2\ R2\ S2\ C5 + S1\ C2\ R2\ S3\ C6 \end{aligned}$$

This program would then proceed to take these eight tie-sets and subtract the intersections taken in pairs. To this it would add the intersections of the tie-sets taken three at a time and subtract the intersections taken four at a time and so forth. The final equation is then simplified and component probability values are substituted for the events, resulting in the success equation.

For our sample problem then the success equation reduces to:

$$P(\text{Success})=S^2\ C^2R(8-4SC-4CR-4SCR+10SC^2R-6SC^3R-40+2SC^2+2C^2R+SC^4R)$$

with the reliability 0.8908 for the engineering data and 0.9327 from the test data.

In this form it is quite simple to vary the values of the components since we are now dealing with a simple algebraic equation. For example, if we wanted to perform an extremely detailed sensitivity study varying each of the component reliability values we could perform the study simply and quickly.

GO

The GO method is basically a "Tree" technique; however, Kaman Sciences Corporation has refined the method to such an extent as to make it an almost totally unique approach. To start with, the program does not merely consider system success and failure but instead defines many time zones either referenced to events in the system operation or equally spaced. Having established a time frame for reference the system operation is described in terms of the transfer of electrical or mechanical actions, called signals, at specific points in the system. Each signal is systematically assigned probabilities of occurrence in each of the time zones in accordance with predetermined rules.

In order to fully understand the operation of the GO program there are several concepts which must be discussed. These include: time zones, signal flow, component types, component kinds, and GO charts. First, we will discuss time zones. Time is divided into a finite number of zones. Typically 8 or 16 zones are used; however, any power of two (2^n) may be used within the limitations of computer storage. These zones are numbered from 0 to 2^n-1. Let us consider 8 zones numbered 0, 1, 2,...7. The final zone, zone 8 would always be used to describe the time of occurrence of events which never occur. Typically that might mean that zone 7 is the time of operation for system "duds" (no-fires). Continuing, the modeler defines zones 1 to 6 as the time periods between major events in the system operation; usually these events are specific inputs. The very first input is usually assigned to time zone 1. From this it follows that time zone 0 (zero) is the time of occurrence of events which occur prior to any inputs in the system. For example a switch may have initially been at an incorrect setting prior to any mission. Thus by utilizing these eight time zones the occurrence of events may be sequenced in time. The next concept to be considered is that of signal flows. The GO program considers the system as a tree of branches and nodes where the nodes are components and the branches are signals. Certain rules are assigned. Once a signal has reached a component it remains available there indefinitely. All signal flow progresses with time, i.e., absolutely no feedback loops are permitted. For the purpose of this program feedback loops are defined by a signal serving as a source of an input to the component which generates that signal. Associated with each signal is a distribution of its probability of occurrence in each time zone. These probabilities are determined by the functioning of the component which produces the signal, and that component's inputs. It is interesting to note that the GO procedure allows the existence of a small error term. That is necessary to limit the total number of branches on the tree. Thus any branch whose total probability falls below a cutoff point (called PMIN) is eliminated from the tree. It is this "pruning" of the tree which allows the program to handle systems which are too complicated for other tree methods.

The next concept of interest is that of "component types." In order to permit the modeling of systems the GO program defines eleven basic "component types." Each component type has a precisely defined set of rules regarding the number of input and output signals, the time relationship of the output to the inputs and probabilistic distribution of the output in terms of the input. In addition, the modeler can determine, as each component is considered, whether its input signals are necessary for retention for further consideration or whether the identity of that input may be eliminated from the branches of the tree. This combining of branches also prevents the tree from becoming unwieldy and shortens the computation time considerably. A complete description of the component operation is available in Reference 4. If any modeling of a real system is to be done, it is imperative that the modeler not only read Reference 4 in its entirety but also review every step in the component logic internal to the program. As a result, only a brief description of each component type is being presented.

Type 1 is a two state component with one input and one output. If the component fails, the output is in time zone 7; otherwise the output is in the time zone of the input.

Type 2 is a perfect "or" gate with two inputs and one output. The output is always in the time zone of the first of the two inputs.

Type 3 is a series chain consisting of an actuator, a normally closed contact (type 7) and a two state (type 1) device. This type is used to represent a three state device with one input and one output. The actuator may either premature, function properly or dud (i.e., fail to operate). The output will be in either time zone 0, 7, or the time of the input signal; depending on the functioning of the three items making up the type 3 component. Any one of these 3 items may be considered perfect if the item is not present in the real system.

Type 4 consists of a parallel pair of normally open contacts (type 6) connected by an "or gate" (type 2). This type is included merely as a convenience since that combination occurs so frequently in real systems. There are three inputs to the type 4 component, these being the common power through each contact and the two actuating signals, one for each signal. One output is produced in any one of five time zones, either zone 0, 7, or the time of one of the three inputs depending on the sequencing of the inputs.

Type 5 is a perfect signal generator. It requires no inputs and produces one output in the desired time zone with probability 1.0. All models must begin with either a type 5 or type 11 component to initiate signal flow.

Type 6 is a normally open contact. It has two inputs and one output signal. Of the two inputs, one is power through the contact while the other actuates contact closure. Outputs may occur when power reaches the contact in the case of a premature function, or after the second of the signals reaches the contact for normal operation. Contact failure precludes an output.

Type 7 is a normally closed contact. It will continue to pass an input power signal until actuated unless the contact prematurely opens.

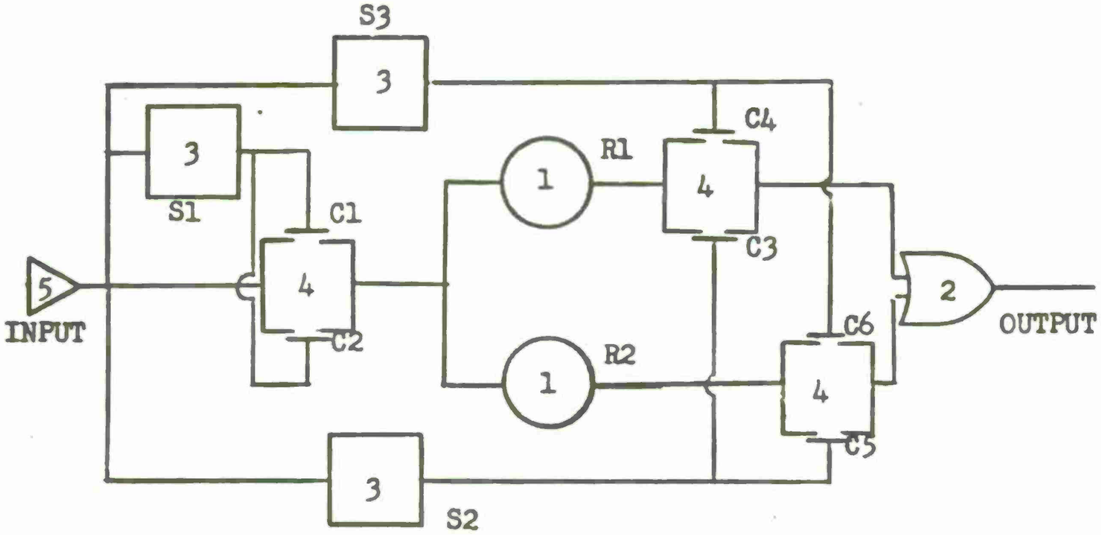
Type 8 is a triggered delay generator. This component has one input and one output. The output occurs a predetermined number of time zones after the input signal is received.

Type 9 is a functional delay generator. There are two inputs, the power and the actuating input. This is a perfect component, i.e., no failures are considered. Output occurs a number N of time zones subsequent to the second signal where N is a function of the interval between the two inputs.

Type 10 is a perfect "and" gate. Output occurs after the second of two inputs is received. This is another perfect component like types 2, 5, 9 and 11.

Type 11 is a stochastic generator. It functions like type 5 except that the output may occur in any time zone according to a probability distribution defined by the user.

Having described the eleven basic component types one last principle remains. That is the construction of a GO chart for the system. To demonstrate this we will return to our sample problem. For our chart we will use the symbols as indicated in the reference report and indicate the type inside the symbol to aid the reader. Our sample problem produces the chart below.



The results for the engineering data are .8909 and for the test data .9327. For this example PMIN was set for zero so there is no error term.

TRI-SERVICE METHOD

The TRI-SERVICE approach is designed to give a distributed estimate of system reliability from test data. While it was not intended to be used for engineering estimates it is also adaptable to this purpose. The actual model for the TRI-SERVICE approach is quite similar to the Boolean success model used in most of the other methods we have looked at. The main differences arise from the assumption that the data assigned to any block in the model is itself descriptive of a distribution and thus the TRI-SERVICE approach utilizes distributed values for the component reliabilities and combines them by a Monte Carlo method to give a distributed estimate of system reliability.

Let us now consider the sample problem. The first step is to construct a model of the system such as that used for the success equations. Actually any of the Boolean models we have looked at would suffice; however, for the sake of discussion we will only look at the success equation model. After constructing this model we then proceed to write the success equation exactly as we have done before. The next step is to assign an initial condition to each block in the system. We will not explore this "prior distribution" as the TRI-SERVICE group referred to it. Suffice it to say that they have a procedure for arbitrarily assigning initial conditions to each block in such a way as to produce results with certain properties. Having assigned prior data to each component this is then updated using the observed data. For attribute (pass/fail) data, each component is described by a beta distribution with the parameter representing successes and failures. The updating of the component data is then merely a process of adding the like parameters of the prior data and the observed data. In the case of our engineering estimates the analyst must assume sample sizes based on his degree of belief in each estimate of component reliability. While this subjectivity is undesirable it is also unavoidable in any "before the fact" analysis of a design. The case of our actual test data is simple since there are observed values for both of the parameters of the component beta distributions. Now to proceed with the sample problem. Having gotten a success equation for the system and a reliability distribution for each component we proceed to perform a Monte Carlo.* From each block in the model a point estimate is randomly selected in accordance with the updated component distribution. These point estimates are then

*Method of moments may also be used.

entered into the success equation as described previously and a system point estimate is calculated. This is repeated a large number of times resulting in a collection of system point estimates. These can then be used to construct a histogram of system reliability estimates. Using this histogram the analyst is then free to make statements about the system reliability including measures of dispersion. Furthermore, he may even approximate the histogram with a beta distribution and make the reliability statements based on this distribution. This allows the interpretation of the parameters of this distribution as equivalent numbers of system successes and failures, an interpretation which has proven useful at times. For the sample system we have estimates of .8904 for the engineer data and .9330 for the actual test data.

SABRE

The SABRE method is a direct offshoot of the TRI-SERVICE approach. The only significant change is that SABRE primarily uses the GO method for calculating the point estimate at each step of the Monte Carlo simulation. Therefore, a detailed description of the routine is unnecessary, the procedure being identical with that described in the past chapter. The principal reason for consideration of SABRE is that unlike the TRI-SERVICE approach, SABRE has achieved acceptance within the Army for use on certain developmental weapons systems. The main reason for the acceptability of the SABRE method is that it utilizes a total approach to the problems of testing and analysis and as such considers a number of engineering problems in the framework of an analytic procedure. Thus in effect, SABRE is a more comprehensive approach to the TRI-SERVICE method.

COMPARISON OF TECHNIQUES

Four of the methods presented thus far are used by agencies involved in evaluation of nuclear weapons reliability. These four, SABRE, GO, SYSTEMEX, and failure equations, are each supported by their users for various reasons. We will compare the methods in four main areas: preparation of the model, performance of the analysis, accuracy of the results, and usefulness of the outputs.

Both SABRE and GO require the preparation of a GO chart prior to performing the analysis as opposed to use of a Boolean success model by the other two methods. The GO chart is generally more difficult to prepare than a Boolean model and is a more time consuming procedure. In addition, it is usually more difficult to check this model than the Boolean model. The GO chart does have certain advantages which outweigh those shortcomings. The model is more detailed and presents a truer representation of the functioning of the system than a Boolean model. The GO model permits proper modeling of component prematures, time dependencies and sequencing of events. In general, more complex intercomponent relationships can be modeled using a GO chart. Thus, SYSTEMEX and failure equations use a simpler, quicker to prepare model while SABRE and GO use a more detailed model more representative of the system.

Without doubt failure equations are the simplest analysis to perform, so much so that a computer is not required for the analysis. On the other extreme SABRE and GO always require a computer and SYSTEMEX usually requires simplification of the model even before a computer can handle the analysis. GO and SABRE are also more difficult to prepare data for program input and must be rerun with each new set of data. Failure equations and SYSTEMEX produce closed form algebraic equations and thus do not require rerunning for new data. This makes them more useful and economical for sensitivity analyses.

As the simple example illustrated all four methods are capable of comparable precision. The accuracy of the results is more dependent on the existence of either gross or subtle errors in the model rather than the calculations which follow. Thus, while an item is in the research and development stage GO or SABRE can give better accuracy by using a more detailed model but once actual flights are performed the data is so gross as to make this detail unwarranted.

Comparing the outputs of the various methods we see that SYSTEMEX and failure equations both provide essentially equal reliability point estimates in the region of interest. The equation produced by the failure approach is much more useful than the success equation from SYSTEMEX. The latter produces an equation so lengthy as to be meaningless to the reviewer while the failure equation is generally a couple of lines at most and easy for a reviewer to cross-check with any model or even from his general knowledge of the particular item. This makes the failure equation the most useful form for reports which will have a generalized distribution among management type engineers. On the other hand GO produces the most useful outputs for the working level engineers and analysts knowledgeable about precisely how the item operates. Finally, SABRE presents data in a manner more useful to program managers, who require distributed estimates of reliability for further analyses of aggregates of many systems as in war gaming or trade-off studies, for example. It is up to the analyst to determine which method produces output suitable for the users of the particular study.

OTHER INVESTIGATIONS

The question of improved methods of reliability analysis has been explored since the mid 1960's. As early as 1968 this writer had explored the various methods then coming into use (Ref 8). That report concluded that none of the methods investigated was an all purpose approach and flexibility should be maintained. In 1971, Mr. P. J. Davitt, Jr. of Picatinny Arsenal published a report (Ref 9) which compared the Monte Carlo method, failure equations and the Kaman GO program. That report recommended the use of both GO and failure equations as a cross-check with the publication of the GO version.

In an early report from Sandia Corporation (Ref 10) the failure and success approaches were compared. That report concluded that failure equations produced equally satisfactory results with much less difficulty than success equations. A later paper (Ref 11) from Sandia Corporation critiqued the GO program and suggested additions, the Sandia view being that GO was not an acceptable routine for reliability analysis. That paper was responded to by Kaman Sciences in 1972 in a report (Ref 12) which detailed many apparent flaws in the Sandia critique. The conclusion of the Kaman report was that their GO routine was a valid technique.

CONCLUSIONS

Four methods for calculating reliability of complex systems have been investigated. Each of these methods is suitable to perform the desired analyses with sufficient accuracy. The failure equation method is always preferable to the success approach and the use of SYSTEMEX should be discontinued. This is based on the fact that the success equation output from SYSTEMEX is much less useful than a failure equation. With the elimination of the SYSTEMEX the other three methods all are useful in certain situations. In early development GO can be used to develop engineering design estimates, and failure equations provide a good cross-check for the model when performed independently. In this case it is preferable to publish the reports using failure equations, which are more universally understandable and meaningful to a reviewer. Finally, in situations where substantial test data has been taken and it is desired to base results on the test data, then SABRE is the most useful approach and gives the most meaningful output results. On the whole, it is desirable to leave the choice of analytical methods to the analyst.

REFERENCES

1. Barnett, Bruce D., "QUEST, A Program for Estimating System Reliability by a Monte Carlo Method," Picatinny Arsenal Report DPSO-IR-6, June 1966.

2. Mardo, John G. and Ricciardi, Anthony J., "A Computerized Procedure for Writing Mathematical Models for Systems Reliability," Picatinny Arsenal Report SMUPA-NR-6-67, February 1967.

3. Loniewski, Edward E., Mardo, John G., et al, "SYSTEMEX, A Computer Program for Reliability Mathematical Modeling," Picatinny Arsenal Report SMUPA-NR-10-68, July 1968.

4. Gately, Wilson T., Stoddard, Dan W., et al, "GO, A Computer Program for the Reliability Analysis of Complex Systems," KAMAN NUCLEAR Report KN-67-704(R), April 1968.

5. Mardo, John G., Cole, Peter V. Z., et al, "A Tri-Service Bayesian Approach to Nuclear Weapons Reliability Assessment," Picatinny Arsenal Report TR-4359, April 1972.

6. Heyderman, Arthur J., and Bernstein, Robert M., "The SABRE Method: Designing a Reliability Test Program for an Artillery Fired Atomic Projectile (AFAP)," Picatinny Arsenal Report 4775.

7. Corneilson, Howard G., "A Computerized Program for Expanding Boolean Equations," Picatinny Arsenal Report SMUPA-NR-15-69, June 1969.

8. Heyderman, Arthur J., "QAD Support for NDED's Safety and Reliability Program," CMT 2 Picatinny Arsenal Internal Correspondence not for public release, March 1968.

9. Davitt, Paul J., "An Evaluation of Three Methods Used in Determining the Reliability of Complex Systems," Picatinny Arsenal Report TM-2009, May 1971.

10. Smith, Richard E., and Goddard, C. H., "Exact Success Equations versus Approximate Dud Equations," Sandia Corporation Internal Correspondence not for public release, December 1968.

11. Anonymous, "Limitations of KAMAN GO Program," Sandia Corporation undated and unsigned.

12. Anonymous, "Response to Sandia Critique of the KAMAN 'GO' Reliability Analysis Method," Kaman Sciences Corp. memo KN-72-625 (M), September 1972.

DISTRIBUTION LIST

Copy No.

Office, Chief of Research & Development 1
Department of the Army
Washington, DC 20310

Commander
U.S. Army Materiel Command
ATTN: AMCQA 2-3
Washington, DC 20315

Commander
U.S. Army Safeguard Systems Office 4
1320 Wilson Boulevard
Arlington, VA 22209

Commander
U.S. Army Safeguard Systems Command
ATTN: SAFSC-Q 5
Huntsville, AL 35890

Commander
U.S. Army Armament Command
ATTN: AMSAR-QA 6
AMSAR-ASF 7
Rock Island, IL 61201

Commander
U.S. Army Combat Developments Command
ATTN: Institute of Nuclear Studies 8
Fort Bliss, TX 79916

Commander
U.S. Army Safeguard Systems Evaluation Agency 9
White Sands Missile Range, NM 88002

Commander
Harry Diamond Laboratories 10
Washington, DC 20438

Commander
U.S. Army Nuclear Weapons Surety Group 11
Fort Belvoir, VA 22060

Commander
Frankford Arsenal
ATTN: SARFA-Q-3000 12
Philadelphia, PA 19137

Commander 13
Edgewood Arsenal
Edgewood, MD 21010

Commander
Watervliet Arsenal
ATTN: SARWV-QA 14
Watervliet, NY 12189

Commander
U.S. Army Aberdeen Research & Development Center
Army Materiel System Analysis Activity
ATTN: Mr. R. Eisner 15
Aberdeen Proving Ground, MD 21005

Director 16
Weapons System Evaluation Group
400 Army Navy Drive
Arlington, VA 22202

Sandia Laboratories
P.O. Box 969
ATTN: Org. 8139 17
Livermore, CA 94550

Commander
Defense Documentation Center 18-29
Cameron Station
Alexandria, VA 22314

Commander
Picatinny Arsenal
ATTN: SARPA-TS-S 30-34
SARPA-AD 35
SARPA-MI 36
SARPA-ND 37
SARPA-QA 38
SARPA-QA-A 39
SARPA-QA-T-C 40-41
SARPA-QA-N 42-51
Dover, NJ 07801